Constraint Back-Offs for Safe, Sufficient Excitation: A General Theory with Application to Experimental Optimization

Gene A. Bunin

Ronin Institute for Independent Scholarship

Abstract

In many experimental settings, one is tasked with obtaining information about certain relationships by applying perturbations to a set of independent variables and noting the changes in the set of dependent ones. While traditional design-of-experiments methods are often well-suited for this, the task becomes significantly more difficult in the presence of constraints, which may make it impossible to sufficiently excite the experimental system without incurring constraint violations. The key contribution of this paper consists in deriving constraint back-off sizes sufficient to guarantee that one can always perturb in a ball of radius δ_e without leaving the constrained space, with δ_e set by the user. Additionally, this result is exploited in the context of experimental optimization to propose a constrained version of G. E. P. Box's evolutionary operation technique. The proposed algorithm is applied to three case studies and is shown to consistently converge to the neighborhood of the optimum without violating constraints.

Keywords: design measures for robustness, evolutionary operation, optimization under uncertainties, experiment design

1. Introduction

In most branches of science, one often encounters systems where the relationship between some experimental response and a finite number of independent variables needs to be studied (Montgomery, 2012; Myers et al., 2009), and it is generally assumed that the response is a dependent variable and a function of the independent ones. Mathematically, the experimental quantity may be stated as the function $f : \mathbb{R}^{n_u} \to \mathbb{R}$, while $\mathbf{u} \in \mathbb{R}^{n_u}$ may be used to denote the vector of independent variables $\mathbf{u} = (u_1, ..., u_{n_u})$. One is then left with the task of identifying $f(\mathbf{u})$. The identification may be either local or global, and is usually done by conducting a series of experiments with different

values of $\mathbf{u}$, observing the resulting $f(\mathbf{u})$ values, and performing some sort of regression. Such procedures are typically used to:

(i) construct data-driven model approximations of f for when f is difficult to model via first principles (Jones et al., 1998; Montgomery, 2012; Myers et al., 2009),

(ii) estimate the uncertain parameters of an already available model (Box, 1990; Chen & Joseph, 1987; Pfaff et al., 2006; Quelhas et al., 2013),

(iii) explore how the function value changes so as to find conditions for which the value is minimized, maximized, or equal to a certain quantity (Box & Wilson, 1951; Conn et al., 2009; Lewis et al., 2000; Robbins & Monro, 1951).

It is the case for many problems that the experimental space of interest is a box defined by the constraints $u_i^L \leq u_i \leq u_i^U$, $i = 1, ..., n_u$, where $\mathbf{u}^L = \left(u_1^L, ..., u_{n_u}^L\right)$ and $\mathbf{u}^U = \left(u_1^U, ..., u_{n_u}^U\right)$ are the lower and upper limits on the independent variables, respectively. Such problems typically correspond to simple set-ups that do not possess major safety limitations, and where testing any variable combination in the experimental space is permissible. Obtaining knowledge about f is not difficult in such conditions, and the traditional design-of-experiments techniques (Montgomery, 2012) are perfectly appropriate here.

However, there still exists a fair share of problems – many of them corresponding to continuous or batch chemical processes (Bunin, 2016) – where additional constraints enter to reduce the experimental space in a nontrivial manner. These constraints may be expressed as the n_g inequalities

$$g_j(\mathbf{u}) \leq 0, \ j = 1, ..., n_g.$$

In some problems, the functions g_j may represent *experimental* relationships that, like f, can only be divined empirically. It often happens that such experimental constraints are safety or economic limitations – they could, for example, represent an upper limit on the temperature in a continuous reactor, or a lower limit on the purity of a batch-produced chemical. Despite the violation of such constraints being highly undesirable, or even dangerous, there currently exists no easy-to-implement, theoretically rigorous method for guaranteeing that the perturbations carried out on the system satisfy these constraints.

Notably, there does exist a fairly established literature on methods that suppose the existence of a parametric model approximation $g_{m,j}(\mathbf{u}, \boldsymbol{\theta}) \approx g_j(\mathbf{u})$, define Θ as the uncertainty set to which $\boldsymbol{\theta}$ belongs, and then attack the problem via probabilistic formulations by ensuring that $g_{m,j}(\mathbf{u}, \boldsymbol{\theta}) \leq 0$ with sufficiently high probability (Kall & Wallace, 1994; Li et al., 2008; Quelhas et al., 2013; Sahinidis, 2004; Zhang et al., 2002). However, while such methods are theoretically just and robust, they suffer from four major practical drawbacks:

(i) the requirement of a parametric model,

2

(ii) the restriction that the uncertainty be parametric, and that Θ be known,

(iii) the computational issues that arise with probabilistic constraints,

(iv) the conservatism that results from the probabilistic constraints reducing the set of admissible $\mathbf{u}$.

Drawback (i) becomes debilitating when the system at hand is difficult to model, while (ii) is more problematic since many employed models are, often by practical requirement, simplifications and thereby prone to *structural* errors (Chachuat et al., 2009). Drawback (iii) is likely to be significant when the models have many decision variables, many uncertain parameters, and are involved. Simplifications, such as linearizing the model with respect to $\boldsymbol{\theta}$ (Zhang et al., 2002), may be used to avoid this, but ultimately come with the loss of rigor that one would expect from an approximation. Finally, (iv) can be extremely problematic when the parametric uncertainty set is large – as may often occur in practice (Li et al., 2008; Quelhas et al., 2013) – since this may limit the perturbation options, with only a small collection of $\mathbf{u}$ being deemed "safe".

The methodology proposed in the present work avoids these difficulties while maintaining the rigor. Taking a model-free, back-off approach, we simplify and generalize the results of Bunin et al. (2014a) to derive positive values, b_j, that, for a given $\mathbf{u}^*$, allow us to state the guarantee

$$g_j(\mathbf{u}^*) \leq -b_j \Rightarrow g_j(\mathbf{u}) \leq 0, \quad \forall \mathbf{u} \in \mathcal{B}_e, \tag{1}$$

where

$$\mathcal{B}_e = \{\mathbf{u} : \|\mathbf{u} - \mathbf{u}^*\|_2 \leq \delta_e\}.$$

Verbally, this means that given a decision-variable set $\mathbf{u}^*$ known to satisfy the constraints with some slack, one is able to provide a guarantee that the entire ball of radius δ_e surrounding $\mathbf{u}^*$ will satisfy the constraints as well, thereby allowing the user to perturb anywhere within this ball without fear of constraint violation. Despite being local, such a result is nevertheless very useful as it allows a high degree of freedom – a ball permitting perturbation sets of any geometry. As will be shown, the value b_j will depend on the local sensitivities of g_j around $\mathbf{u}^*$, but can nevertheless be computed without requiring much effort from the user. Conversely, δ_e is the *sole tuning parameter* set by the user and represents, in some sense, the magnitude of perturbation considered as "sufficiently exciting" for identification given the particular problem.

To date, this result has already been integrated into the SCFO experimental optimization solver (Bunin, 2015), where it is used to ensure accurate linear and quadratic regression, but it is expected that the generality of the result make it applicable to many algorithms and contexts. In this paper, its usefulness is illustrated for a much simpler optimization algorithm – the evolutionary operation (EVOP) method of Box (Box, 1957; Box & Draper, 1969). As the original method searches to maximize an experimental function by perturbing

3

in a hypercube around the best known $\mathbf{u}$, it is made coherent with the result here by ensuring that the cube lie inside $\mathcal{B}_e$, with $\mathbf{u}^*$ then defined as the best known reference point. By forcing $\mathbf{u}^*$ to always satisfy (1), it thus follows that all exploration by the modified EVOP version satisfy the constraints.

The remainder of this paper is organized as follows. The required mathematical concepts and the derivation of the appropriate constraint back-offs are presented in Section 2. Section 3 then provides a robust extension of (1) that accounts for noise/error in the function values, together with a general discussion of potential implementation issues. The constrained EVOP algorithm is presented in Section 4, and its effectiveness is illustrated for three case-study problems. Section 5 concludes the paper.

2. Derivation of Sufficient Back-Offs

So as to keep the forthcoming analysis relatively simple, the following assumption on the continuity and differentiability of g_j is made.

Assumption 1. *The functions g_j are continuously differentiable ($\mathcal{C}^1$) on an open set containing $\mathcal{B}_e$.*

This then allows for the definition of bounds on the sensitivities of g_j.

Definition 1. *The local Lipschitz constants of g_j are defined as any constants κ_{ji} satisfying*

$$ -\kappa_{ji} \leq \left.\frac{\partial g_j}{\partial u_i}\right|_{\mathbf{u}} \leq \kappa_{ji}, \ \forall \mathbf{u} \in \mathcal{B}_e. \tag{2}$$

The existence of these constants follows from Assumption 1 and the boundedness of $\mathcal{B}_e$. They may be used to bound the violation of a given g_j via the local *Lipschitz upper bound*.

Lemma 1. *Let $\mathbf{u}_a, \mathbf{u}_b \in \mathcal{B}_e$. It follows that*

$$ g_j(\mathbf{u}_b) \leq g_j(\mathbf{u}_a) + \sum_{i=1}^{n_u} \kappa_{ji} |u_{b,i} - u_{a,i}|. \tag{3}$$

PROOF. See Bunin et al. (2014b). □

Finally, the Lipschitz bound may be exploited by substituting $\mathbf{u}_a \to \mathbf{u}^*$ and $\mathbf{u}_b \to \mathbf{u}$ in (3) to generate a *Lipschitz polytope* around the point $\mathbf{u}^*$.

Definition 2. *Let $\mathcal{L}_j$ denote the Lipschitz polytope of the constraint g_j around $\mathbf{u}^*$, defined as the set*

$$ \mathcal{L}_j = \left\{ \mathbf{u} : g_j(\mathbf{u}^*) + \sum_{i=1}^{n_u} \kappa_{ji} |u_i - u_i^*| \leq 0 \right\}. \tag{4}$$

4

The Lipschitz polytope has two important properties that should be apparent by inspection:

(i) $\mathbf{u} \in \mathcal{L}_j \cap \mathcal{B}_e \Rightarrow g_j(\mathbf{u}) \leq 0$,

(ii) $g_j(\mathbf{u}^*) \leq 0 \Rightarrow \mathcal{L}_j \neq \varnothing$.

The "double membership" of $\mathbf{u} \in \mathcal{L}_j$ and $\mathbf{u} \in \mathcal{B}_e$ in (i) is required to ensure both that $\mathbf{u}$ satisfies the upper bound of (3) and that the bound itself is valid to begin with, respectively. Property (ii) should be evident if one just considers $\mathbf{u} := \mathbf{u}^*$ when $g_j(\mathbf{u}^*) \leq 0$.

Furthermore, it is clear that the content (hypervolume) of $\mathcal{L}_j$ increases monotonically as $g_j(\mathbf{u}^*)$ decreases – i.e., $\mathcal{L}_j$ admits more and more implementable points because of the terms $|u_i - u_i^*|$ being allowed to grow larger while satisfying the inequality.

It is this observation that inspires the foundations of the present work, illustrated geometrically in Figure 1. If $g_j(\mathbf{u}^*)$ can be forced to remain sufficiently low – i.e., if $g_j(\mathbf{u}^*)$ can be made to satisfy the constraint with a certain back-off – then one can always guarantee the existence of a non-empty Lipschitz polytope centered at $\mathbf{u}^*$. Additionally, because the ball $\mathcal{B}_e$ is also centered at $\mathbf{u}^*$, it may be inscribed inside the polytope, with Property (i) above then sufficient to guarantee that all points in this ball satisfy the constraint $g_j(\mathbf{u}) \leq 0$.

The required size of this back-off is easily derived by exploiting the Cauchy-Schwarz inequality.

Theorem 1. *Let $\boldsymbol{\kappa}_j$ denote the vector $(\kappa_{j1}, ..., \kappa_{jn_u})$ corresponding to a given δ_e and $\mathbf{u}^*$. If $b_j \geq \delta_e \|\boldsymbol{\kappa}_j\|_2$, then the implication (1) holds.*

PROOF. Defining the quantity $\Delta u_i = |u_i - u_i^*|$ and the vector $\Delta \mathbf{u} = (\Delta u_1, \ldots, \Delta u_{n_u})$, let us restate the definition of the Lipschitz polytope as

$$\mathcal{L}_j = \left\{ \mathbf{u} : g_j(\mathbf{u}^*) + \boldsymbol{\kappa}_j^T \Delta \mathbf{u} \leq 0 \right\}.$$

From the Cauchy-Schwarz inequality, it follows that

$$\boldsymbol{\kappa}_j^T \Delta \mathbf{u} \leq \|\boldsymbol{\kappa}_j\|_2 \|\Delta \mathbf{u}\|_2 = \|\boldsymbol{\kappa}_j\|_2 \|\mathbf{u} - \mathbf{u}^*\|_2, \qquad (5)$$

which then allows us to consider a (ball) subset of $\mathcal{L}_j$:

$$\hat{\mathcal{L}}_j = \left\{ \mathbf{u} : g_j(\mathbf{u}^*) + \|\boldsymbol{\kappa}_j\|_2 \|\mathbf{u} - \mathbf{u}^*\|_2 \leq 0 \right\},$$

where $\hat{\mathcal{L}}_j \subseteq \mathcal{L}_j$ is evident since any $\mathbf{u}$ satisfying

$$g_j(\mathbf{u}^*) + \|\boldsymbol{\kappa}_j\|_2 \|\mathbf{u} - \mathbf{u}^*\|_2 \leq 0$$

must also satisfy

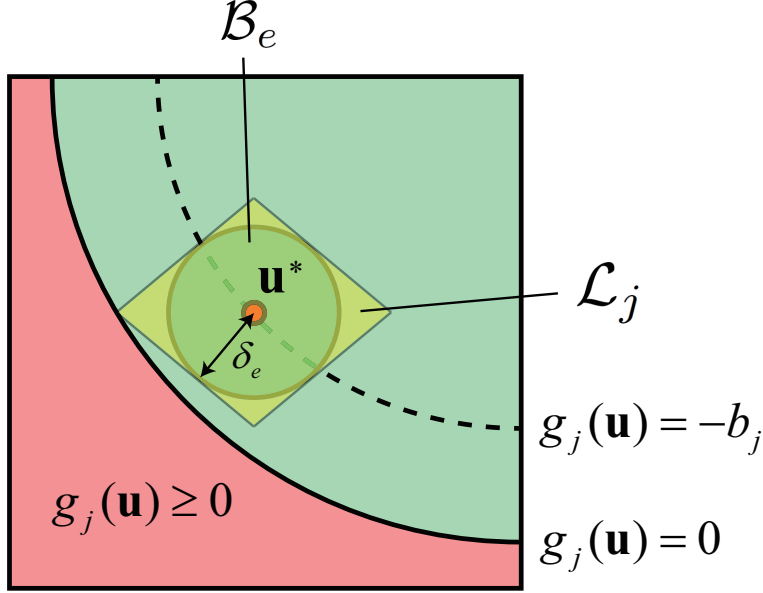$$g_j(\mathbf{u}^*) + \boldsymbol{\kappa}_j^T \Delta \mathbf{u} \leq 0$$

Figure 1: Geometric illustration of the concept behind using back-offs to enforce the existence of an excitation ball $\mathcal{B}_e$. The back-off is chosen large enough so that a ball of radius δ_e may be *inscribed* inside the Lipschitz polytope corresponding to the back-off.

by virtue of (5).

It is then sufficient to show that $\mathcal{B}_e \subseteq \hat{\mathcal{L}}_j$ for the back-off specified. Because every $\mathbf{u} \in \mathcal{B}_e$ satisfies $\|\mathbf{u} - \mathbf{u}^*\|_2 \leq \delta_e$, it follows that any point in $\mathcal{B}_e$ will belong to $\hat{\mathcal{L}}_j$ if the inequality $g_j(\mathbf{u}^*) + \delta_e \|\boldsymbol{\kappa}_j\|_2 \leq 0$ holds, which is precisely what is ensured by the back-off. Thus, $\mathcal{B}_e \subseteq \hat{\mathcal{L}}_j \subseteq \mathcal{L}_j$, with Property (i) of the Lipschitz polytope then leading to the desired result. $\qquad\square$

It is natural to ask what one should do if, given δ_e, $\mathbf{u}^*$, and (consequently) κ_{ji}, the condition $g_j(\mathbf{u}^*) \leq -\delta_e \|\boldsymbol{\kappa}_j\|_2$ does not hold. One solution consists in choosing a different, safer $\mathbf{u}^*$ from the available measurements, maintaining the same δ_e, and hoping that the back-off is satisfied for this new choice. Alternatively, one may keep $\mathbf{u}^*$ the same and reduce δ_e. Assuming that $g_j(\mathbf{u}^*) < 0$, it is not hard to show that for δ_e sufficiently small the back-off will be met. Consider, for example, the reduction sequence

$$\left\{ \frac{\delta_e}{2^n} \right\}, \ n \in \mathbb{N} = \{0, 1, 2, \dots\},$$

together with the corresponding $\mathcal{B}_e$:

$$\left\{ \mathbf{u} : \|\mathbf{u} - \mathbf{u}^*\|_2 \leq \frac{\delta_e}{2^n} \right\}.$$

Clearly, each reduced $\mathcal{B}_e$ will be a strict subset of the previous, from which we may deduce that the κ_{ji} valid for $n := 0$ will remain valid for all n, thus allowing us to upper bound the $\|\boldsymbol{\kappa}_j\|_2$ portion of the back-off and to conclude that $\delta_e \|\boldsymbol{\kappa}_j\|_2$ may be made arbitrarily close to 0 by choosing n sufficiently large (δ_e sufficiently small), thereby resulting in a back-off that is satisfied by $g_j(\mathbf{u}^*)$.

3. Implementation Issues

On the surface, the derived result seems easy to implement as it simply states that if, given $\mathbf{u}^*$, one wants to perturb anywhere in $\mathcal{B}_e$ without fear of constraint violation, one only needs to confirm that $g_j(\mathbf{u}^*) \leq -\delta_e \|\boldsymbol{\kappa}_j\|_2$, $\forall j$. If this condition does not hold, one can then choose a different $\mathbf{u}^*$ from the set of applied $\mathbf{u}$ or decrease δ_e. However, there remain a number of implementation aspects that merit discussion.

3.1. Setting of Lipschitz Constants

Theorem 1 provides a robust guarantee only if it is assumed that the Lipschitz constants provided satisfy (2). While one can argue that picking very large, conservative values is sufficient, this has the obvious performance drawback of increasing $\delta_e \|\boldsymbol{\kappa}_j\|_2$ and thus b_j, which could make satisfying $g_j(\mathbf{u}^*) \leq -b_j$ impossible. As such, one would prefer setting these constants in more intelligent ways.

Apart from some scarce and limited results in the global optimization literature (Hansen et al., 1992; Strongin, 1973; Wood & Zhang, 1996), the problem of estimating Lipschitz constants effectively is still very open. Some techniques have been proposed in the recently submitted manuscript of Bunin & François (2016), and suggest mixing *a priori* knowledge about the functions g_j, which may provide good initial guesses of the constants, together with data-driven refinements, making explicit use of the fact that for a sufficiently local or approximately linear region the Lipschitz constants differ little from the local function derivatives. In the case that parametric models $g_{m,j}(\mathbf{u}, \boldsymbol{\theta})$ are available, one could bolster the *a priori* knowledge with the robust model-based estimates

$$\kappa_{ji} := \max_{\substack{\mathbf{u} \in \mathcal{B}_e \\ \boldsymbol{\theta} \in \Theta}} \left| \frac{\partial g_{m,j}}{\partial u_i}\Big|_{\mathbf{u}, \boldsymbol{\theta}} \right|$$

as the Lipschitz-constant estimates. This mix of techniques has been employed in the SCFO solver (Bunin, 2015) and has generally led to relatively robust performance in test problems, although constraint violations may still occur from time to time (Bunin, 2016).

It is worth noting that the EVOP method proposed in Section 4 appears to handle this issue *very well*, using iterative local linear regression to estimate the Lipschitz constants in a manner that requires no input from the user and yet avoids constraint violation completely.

3.2. Accounting for Noise/Error

Enforcing $g_j(\mathbf{u}^*) \leq -b_j$ also relies on having accurate knowledge of $g_j(\mathbf{u}^*)$. While it is reasonable to expect that in most applications one will be able to either measure or estimate the values of these functions at $\mathbf{u}^*$, for experimental functions it is generally the case that a perfect measurement or estimation will be impossible, and that there will only be access to the corrupted values, $\hat{g}_j^*$. A typical assumption (Hotelling, 1941; Marchetti et al., 2010; Moré & Wild, 2011; Myers et al., 2009) is that this corruption is additive – i.e., that

$$\hat{g}_j^* = g_j(\mathbf{u}^*) + w,$$

with w a stochastic element for which the estimates of at least the mean and variance are available. In this case, it becomes possible to compute a high-probability bounding value, $g_j(\mathbf{u}^*) \leq \overline{g}_j^*$, using, in the most general case, Chebyshev's inequality (Moré & Wild, 2011), or something less conservative if better assumptions on the nature of the noise are available. More involved techniques for computing $\overline{g}_j^*$ are outlined in Section 4 of Bunin et al. (2014a) and in the recent work of Bunin & François (2016).

One may then work with the robust condition $\overline{g}_j^* \leq -b_j$ instead, since satisfaction of this condition implies the satisfaction of $g_j(\mathbf{u}^*) \leq -b_j$ with a high probability.

3.3. Accomodating Numerical Constraints

So far, g_j has been treated from a very general perspective, and has been assumed only to be a $\mathcal{C}^1$ function over $\mathcal{B}_e$. However, there are plenty of constraints for which *much more* knowledge is available. In particular, when the constraint g_j is a numerical function that can be evaluated by hand or by computer for any desired $\mathbf{u} \in \mathcal{B}_e$, the guarantee of perturbing in $\mathcal{B}_e$ without incurring violation of the given constraint is fairly easy. Namely, it suffices to ensure that

$$\max_{\mathbf{u} \in \mathcal{B}_e} g_j(\mathbf{u}) \leq 0,$$

since this trivially guarantees that all perturbations in $\mathcal{B}_e$ are feasible for that constraint, without requiring Lipschitz constants or robust upper bounds. For some cases, it may be that g_j is an involved *nonconcave* function, potentially making its maximization over a ball a numerically challenging (global optimization) task. When this occurs, one could attempt to take a concave upper bound or, if worse comes to worst, estimate its Lipschitz constants – a significantly easier task for a numerical function – and then employ the general (albeit conservative) result of Theorem 1.

A special subclass of numerical constraints that is even easier to work with is that of the bound constraints, $u_i^L \leq u_i \leq u_i^U$, which are extremely relevant since they tend to occur in virtually any well-defined experimental investigation. By simple intuition and quick inspection, one should be able to see that employing a back-off of δ_e is sufficient for these bounds. Not surprisingly, it is possible to

arrive at this same result by Theorem 1, as the $\boldsymbol{\kappa}_j$ vector for these constraints may simply be taken as the vector of $n_u - 1$ zeros with unity in the ith spot, thus leading to $\delta_e \|\boldsymbol{\kappa}_j\|_2 = \delta_e$.

One could, however, choose not to back off from the bound constraints since, given their orthogonal nature, one is always guaranteed to retain a full orthant in which one can perturb. Given with the feasibility of $\mathcal{B}_e$, this means that one could ignore these back-offs and still retain $1/2^{n_u}$ of $\mathcal{B}_e$ for safe perturbation. For most cases, this fraction of $\mathcal{B}_e$ still offers enough room for the user to accomplish what they aim to, such as estimating a derivative or locally exploring the function's behavior.

3.4. Scaling and the Tuning of the Excitation Radius

It is evident that the results provided are not scale-invariant, and that a ball of radius δ_e could offer very poor perturbation in directions that vary over a much greater domain than others – geometrically, this may be seen as the problem of the efficiency of inscribing a ball inside a Lipschitz polytope. One solution could be to generalize the notion of the ball to an ellipse, and to rederive the result for this more versatile case. However, a simple approach found to work well has been to simply scale the independent variables so that each u_i varies between 0 and 1, in which case perturbing in a ball is reasonable.

This aside, one may still have to make a nontrivial decision about what δ_e to set. Ultimately, this choice should depend on the particular problem that the user is trying to solve, with all available *a priori* knowledge exploited to yield an appropriate choice. For example, if the goal is to explore as much of the experimental space as possible and to build a model that encompasses a large, perhaps even global, domain, it may be of interest to set δ_e to be as large as possible. If the goal is to perturb just enough to estimate a derivative while avoiding significant corruption due to noise, one may want to choose δ_e as a function of the expected signal-to-noise ratio (Bunin, 2015, §3.14). When information about the function's curvature is available, it may be possible to choose δ_e so as to strike a balance between the corruption due to noise and the corruption due to nonlinearity (Marchetti et al., 2010). Finally, the *ad hoc*, post-scaling choices of $\delta_e := 0.01, 0.05, 0.10$ have also often worked well, in the author's experience.

4. Feasible-Side Evolutionary Operation

An evolutionary operation (EVOP) algorithm is constructed for the feasible-side solution of experimental optimization problems having the form

$$
\begin{aligned}
\underset{\mathbf{u}}{\text{minimize}} \quad & \phi(\mathbf{u}) \\
\text{subject to} \quad & g_j(\mathbf{u}) \leq 0, \qquad j = 1, ..., n_g \\
& u_i^L \leq u_i \leq u_i^U, \quad i = 1, ..., n_u,
\end{aligned}
\tag{6}
$$

where both ϕ and g_j will, in general, represent experimental functions. As in many experimental optimization problems of this type, there is a constant

9

trade-off between adapting the decision variables $\mathbf{u}$ so as to minimize ϕ and perturbing them so as to learn more about the local or general natures of ϕ and g_j. Both tasks are necessary – one cannot optimize without perturbation, but one cannot optimize if one spends all of the experimental resources perturbing the system for knowledge, either.

Two main reasons motivate this choice of application. First, the derived result of Theorem 1 is innately coherent with the direct-search algorithmic nature of EVOP, which iteratively perturbs in a local region and then shifts this region so that it is centered around the best found point. It is thus quite natural to let EVOP perturb in $\mathcal{B}_e$ and then shift $\mathbf{u}^*$ accordingly, which immediately provides the traditional EVOP with constraint satisfaction guarantees. The second reason has to do with the practical usefulness of the constructed scheme. Ever since its inception in 1957 (Box, 1957), EVOP has enjoyed great popularity in industry, largely because of its extreme simplicity. In addition to being simple, it would later be shown to be theoretically well founded as well, with the slight addition of a step-size rule allowing it to enjoy the global convergence properties of Torczon's generalized pattern search algorithms (Torczon, 1997). However, both its industrial and theoretical success has, for the most part, been limited to simple problems with bound constraints, and while one could use the standard (i.e., penalty-function) methods (Conn et al., 2000, Ch. 14) (Fletcher, 1987, Ch. 12) to convert problems with g_j constraints into the bound-constrained form for which EVOP is readily applicable (Rutten et al., 2015), these approaches are ultimately not safe and can at most only offer the guarantee that the constraints are satisfied upon convergence.

From this point of view, the algorithm proposed in this section is believed to carry great potential as a stand-alone contribution that, in addition to being effective for the problems tested, is also extremely easy to apply and requires minimal input from the user.

4.1. Description of the Algorithm

Prior to stating the algorithm, let us first make some knowledge assumptions so as to ensure its basic functionality.

First, it is assumed that, for a given tested $\overline{\mathbf{u}}$, one measures both ϕ and g_j with additive white Gaussian noise:

$$\hat{\phi} = \phi(\overline{\mathbf{u}}) + w_\phi, \quad w_\phi \sim \mathcal{N}(0, \sigma_\phi^2)$$
$$\hat{g}_j = g_j(\overline{\mathbf{u}}) + w_j, \quad w_j \sim \mathcal{N}(0, \sigma_j^2).$$

For simplicity, it will be assumed that σ_ϕ and $\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_{n_g})$ are known.

It will also be assumed that a sufficiently "safe" initial $\mathbf{u}^*$ is available, so that the $2n_u$ test points generated by perturbing the individual elements of $\mathbf{u}^*$ by $\pm\delta_e$,

$$(u_1^* \pm \delta_e, ..., u_{n_u}^*),$$
$$(u_1^*, u_2^* \pm \delta_e, ..., u_{n_u}^*),$$

$$\vdots$$

$$(u_1^*, u_2^*, ..., u_{n_u}^* \pm \delta_e),$$

are, together with $\mathbf{u}^*$, safe and satisfy the constraints.

Scaled variables will be denoted with $(\tilde{\cdot})$ and defined as

$$\tilde{u}_i = \frac{u_i - u_i^L}{u_i^U - u_i^L}, \quad i = 1, ..., n_u.$$

This scaling step will be implicit in the algorithm statement that follows – i.e., we will switch between the scaled and unscaled variables as needed without explicitly including the affine operation above in the algorithm steps.

Algorithm 1 (Feasible-Side EVOP)

1. (Initialization) The initial reference point $\mathbf{u}^*$, the standard deviations σ_ϕ and $\boldsymbol{\sigma}$, and the decision-variable bounds $\mathbf{u}^L$, $\mathbf{u}^U$ are provided. The excitation radius $0.5 \geq \delta_e > 0$ is set by the user.

2. (Evaluation at Reference Point) Apply $\mathbf{u}^*$ to the experimental system to obtain $\hat{\phi}^*$ and $\hat{g}_j^*$. Define $\tilde{\mathbf{U}} := (\tilde{\mathbf{u}}^*)^T$, $\boldsymbol{\phi} := \hat{\phi}^*$, $\mathbf{G} := [\hat{g}_1^* \ \cdots \ \hat{g}_{n_g}^*]$.

3. (Perturbation) For $i = 1, ..., n_u$:

   (a) (Perturbing by $\pm\delta_e$) Obtain $\tilde{\mathbf{u}}_+$ and $\tilde{\mathbf{u}}_-$ by perturbing the ith element of $\tilde{\mathbf{u}}^*$ by δ_e and $-\delta_e$, respectively.

   (b) (Upper Bound Constraint Check) If $u_{+,i} \leq u_i^U$, apply $\mathbf{u}_+$ to the experimental system to obtain the corresponding measurements $\hat{\phi}$ and $\hat{g}_j$, denoted by $\hat{\phi}^+$ and $\hat{g}_j^+$, and augment $\tilde{\mathbf{U}}$, $\boldsymbol{\phi}$, and $\mathbf{G}$:

   $$\tilde{\mathbf{U}} := \begin{bmatrix} \tilde{\mathbf{U}} \\ \tilde{\mathbf{u}}_+^T \end{bmatrix}, \quad \boldsymbol{\phi} := \begin{bmatrix} \boldsymbol{\phi} \\ \hat{\phi}^+ \end{bmatrix},$$

   $$\mathbf{G} := \begin{bmatrix} \mathbf{G} \\ \hat{g}_1^+ \ \cdots \ \hat{g}_{n_g}^+ \end{bmatrix}.$$

   (c) (Lower Bound Constraint Check) If $u_{-,i} \geq u_i^L$, apply $\mathbf{u}_-$ to the experimental system to obtain the corresponding measurements $\hat{\phi}$ and $\hat{g}_j$, denoted by $\hat{\phi}^-$ and $\hat{g}_j^-$, and augment $\tilde{\mathbf{U}}$, $\boldsymbol{\phi}$, and $\mathbf{G}$:

   $$\tilde{\mathbf{U}} := \begin{bmatrix} \tilde{\mathbf{U}} \\ \tilde{\mathbf{u}}_-^T \end{bmatrix}, \quad \boldsymbol{\phi} := \begin{bmatrix} \boldsymbol{\phi} \\ \hat{\phi}^- \end{bmatrix},$$

   $$\mathbf{G} := \begin{bmatrix} \mathbf{G} \\ \hat{g}_1^- \ \cdots \ \hat{g}_{n_g}^- \end{bmatrix}.$$

(d) (Set Number of Points Tested) If

$$0 \le \tilde{u}_i^* \pm \delta_e \le 1,$$

set $s_i := 2$. Otherwise, set $s_i := 1$.

4. (Local Linear Regression) Fit a linear model,

$$\beta_0 + \sum_{i=1}^{n_u} \beta_i \tilde{u}_i,$$

to the observed cost function values, with the $\beta_1, ..., \beta_{n_u}$ coefficients recovered as the first n_u elements of the least-squares solution $[\tilde{\mathbf{U}} \ \mathbf{1}]^\dagger \boldsymbol{\phi}$. Set $\nabla\hat{\phi}^* = (\beta_1, ..., \beta_{n_u})$ as the estimate of the gradient of ϕ at $\tilde{\mathbf{u}}^*$. Repeat the same procedure with the columns of $\mathbf{G}$ to obtain the gradient estimates $\nabla\hat{g}_j^*$ for $j = 1, ..., n_g$.

5. (Lipschitz Constants) For each $i = 1, ..., n_u$ and $j = 1, ..., n_g$, set

$$\kappa_{ji} := \left| \frac{\partial\hat{g}_j^*}{\partial\tilde{u}_i} \right| + 6\frac{\sigma_j\sqrt{2}}{s_i\delta_e},$$

where $\partial\hat{g}_j^*/\partial\tilde{u}_i$ denotes the ith element of $\nabla\hat{g}_j^*$.

6. (Nearly Active Constraints) Define the index set

$$j_A := \left\{ \begin{array}{l} j : \hat{g}_j + 3\sigma_j \ge -\delta_e \|\boldsymbol{\kappa}_j\|_2 \\ \text{for at least one measurement } \hat{g}_j \text{ in the } j^{\text{th}} \text{ column of } \mathbf{G} \end{array} \right\}$$

as the index set of nearly active constraints.

7. (Approximation of Lagrangian) Defining the gradient of the Lagrangian at $\tilde{\mathbf{u}}^*$ as

$$\nabla L(\boldsymbol{\lambda}) = \nabla\hat{\phi}^* + \sum_{j=1}^{n_g} \lambda_j \nabla\hat{g}_j^*,$$

approximate the corresponding Lagrange multipliers, $\boldsymbol{\lambda}^*$, by the solution to the constrained least-squares problem

$$\begin{aligned} \boldsymbol{\lambda}^* := \arg\underset{\boldsymbol{\lambda}}{\text{minimize}} \quad & \nabla L(\boldsymbol{\lambda})^T \nabla L(\boldsymbol{\lambda}) \\ \text{subject to} \quad & \lambda_j \ge 0, \ \ \forall j \\ & \lambda_j = 0, \ \ \forall j \notin j_A. \end{aligned}$$

8. (Choose New Reference) Set as $\tilde{\mathbf{u}}^*$ the point in $\tilde{\mathbf{U}}$ that

    (i) has the smallest $\nabla L(\boldsymbol{\lambda}^*)^T \tilde{\mathbf{u}}$ value,

    (ii) has the corresponding upper bounds $\overline{g}_j^* := \hat{g}_j^* + 3\sigma_j$ that satisfy $\overline{g}_j^* \leq -\delta_e \|\boldsymbol{\kappa}_j\|_2, \ \forall j$.

If no such $\tilde{\mathbf{u}}^*$ exists, maintain the same reference point as before. Define $\tilde{\mathbf{U}} := (\tilde{\mathbf{u}}^*)^T$, $\boldsymbol{\phi} := \hat{\phi}^*$, $\mathbf{G} := [\hat{g}_1^* \ \cdots \ \hat{g}_{n_g}^*]$ and return to Step 3.

A number of remarks are in order:

- As the scaling reduces the variable space to a unit hypercube, there is no need for δ_e to be set above 0.5, as this would preclude the existence of a feasible $\mathcal{B}_e$.

- The perturbation scheme used here differs from that of the traditional EVOP methods, which use a 2^{n_u} factorial scheme to define the test points (Box, 1957; Box & Draper, 1969), and is in this sense more similar to a coordinate search (Conn et al., 2009, Ch. 7). The rationale for doing this is that requiring $2n_u$ perturbations leads to better efficiency as n_u increases (linear as opposed to geometrical), requires fewer perturbations for $n_u > 2$, and appears to be sufficient for the given tasks.

- As suggested in Section 3.3, no back-offs are used for the bound constraints, ensuring instead that they are never violated during the perturbation phase.

- It is not difficult to show that the derivative estimates obtained by linear regression of the local data set are, as a result of the orthogonality and symmetry of the perturbations, identical to the difference quotients

$$\frac{\partial \hat{g}_j^*}{\partial \tilde{u}_i} = \frac{\hat{g}_j^+ - \hat{g}_j^-}{2\delta_e}$$

when $s_i := 2$, or to either

$$\frac{\hat{g}_j^+ - \hat{g}_j^*}{\delta_e} \text{ or } \frac{\hat{g}_j^* - \hat{g}_j^-}{\delta_e}$$

when $s_i := 1$. It is well known (see, e.g., Daniel & Heerema (1950)) that the standard deviation for such estimates is equal to $\sigma_j \sqrt{2}/(s_i \delta_e)$. In adding six times this quantity to the absolute value of the derivative estimates, we use half of that amount ("three sigma") to account for potential errors in the estimates due to noise, and the other half to compensate for the fact that the Lipschitz constants only match the derivatives very locally, and may actually be larger for the $\mathcal{B}_e$ at the future cycle (i.e., the Lipschitz-constants estimation is always one cycle behind). This is important since the new reference is always chosen subject to the

restriction that the corresponding $\overline{g}_j^* \leq -\delta_e \|\boldsymbol{\kappa}_j\|_2$, $\forall j$, where the back-off is defined using the Lipschitz constants at the *current cycle*. This then encourages the guarantee of safe excitation around the new reference.

- Adding $3\sigma_j$ to the constraint measurements in Steps 6 and 8 robustifies the scheme against measurement noise, resulting in the upper bounding values $\overline{g}_j := \hat{g}_j + 3\sigma_j$ that bound the true function values with a high probability of about 99.85%.

- Because the problem addressed is constrained and uses Lipschitz bounds to guarantee robust constraint satisfaction, it follows that the scheme could converge suboptimally if allowed to get too close to a constraint. This issue was first observed by Bunin et al. (2011) and dealt with rigorously by Bunin et al. (2014c), and represents a main drawback of using Lipschitz constants for constraint satisfaction. So as to avoid this issue here, it has been proposed to minimize an approximation of the Lagrangian instead of simply minimizing ϕ. Note that when $\mathbf{u}^*$ is far from any constraints, the scheme simply sets $\boldsymbol{\lambda}^* := \mathbf{0}$ and the two objectives are equal. However, when some g_j start to get close to activity, the scheme defines the objective as a trade-off between minimizing ϕ and lowering the nearly active constraint values, which tends to allow the algorithm to "slide off" the nearly active constraints in application. The theoretical rigor of this scheme is not entirely clear, but it may be seen to be consistent upon convergence, in that finding $\boldsymbol{\lambda}^*$ such that $\nabla L(\boldsymbol{\lambda}^*) \approx \mathbf{0}$ implies that $\mathbf{u}^*$ is a constrained stationary point, provided that the gradient estimates are not too erroneous.

4.2. Application to Case-Study Examples

Three problems are chosen from the ExpOpt database of test problems (Bunin, 2016, Problems P2, P3, P6) so as to illustrate the performance of the algorithm when applied to realistic case-study scenarios. The first problem consists in varying the feed rate and temperature of a Williams-Otto plant so as to maximize its steady-state profit while honoring an upper limit on a noxious product, and is originally taken from the paper of Marchetti (2013). The second, adapted from the works of Gentric et al. (1999) and François et al. (2005), deals with minimizing the operating time of a polysterene batch reactor while honoring a minimal molecular weight specification. Here, two "switching times" that help define the temperature profile of the batch are taken as the decision variables. Finally, the third problem comes from François & Bonvin (2013) and seeks to maximize the steady-state production of a continuous reactor by varying two feed rates subject to two experimental constraints. Two-dimensional problems have intentionally been chosen as they allow for easy visualization and interpretation of the results. The different problem specifications are provided in Table 1.

Inputting the specifications into Algorithm 1, the algorithm's performance is tested for the *ad hoc* choice of $\delta_e := 0.05$. The results are given in Figure 2

Table 1: Problem specifications for the case-study examples.

#	Initial $\mathbf{u}^*$	σ_ϕ	$\boldsymbol{\sigma}$	$\mathbf{u}^L$	$\mathbf{u}^U$
P2	$(3.5, 72)$	0.5	$5 \cdot 10^{-4}$	$(3, 70)$	$(6, 100)$
P3	$(242, 945)$	60	10^4	$(50, 600)$	$(450, 1000)$
P6	$(14.5, 14.9)$	0.1	$(0.03, 0.03)$	$(1, 1)$	$(50, 50)$

and show that the algorithm succeeds in exploring the decision-variable space while satisfying the constraints – not a single violation is observed. By contrast, suppose that one tried to apply the same algorithm to Problem P2 but without the application of the safety back-off derived in Theorem 1. This is done by substituting $-\delta_e \|\boldsymbol{\kappa}_j\|_2 \to 0$ in Steps 6 and 8 of Algorithm 1. The result is given in Figure 3. Again, it is seen that the use of the Lagrangian leads to the algorithm successfully converging to a neighborhood of the optimum. However, constraint violations occur repeatedly along the convergence trajectory.

The convergence properties of the algorithm are of interest – in particular, one notices that the algorithm converges to a region that is relatively suboptimal in Problem P6. This is due to the geometry of the problem, the conservatism of the back-offs, and the size of the noise in the constraint measurements. In other words, the algorithm cannot make further progress while robustly (to $3\sigma_j$) satisfying the specified back-offs. While proving convergence to an optimum in a mathematically rigorous manner is outside the scope of this paper, for $\mathcal{C}^1$ functions one should expect the algorithm to converge properly if both the noise and back-offs go to 0 asymptotically, as $\delta_e \downarrow 0$ would lead to the linear regression converging to the true function gradient in the absence of noise (Conn et al., 2009, §2.4), with the removal of noise also removing the conservatism introduced by using the upper bounds $\hat{g}_j + 3\sigma_j$. To test this hypothesis empirically, let us use a modified implementation of Algorithm 1 where

$$\delta_e(k) := \delta_e/\sqrt{k},$$
$$\sigma_\phi(k) := \sigma_\phi/\sqrt{k},$$
$$\boldsymbol{\sigma}(k) := \boldsymbol{\sigma}/\sqrt{k},$$

with k starting at 1 and being incremented by 1 after each cycle of the algorithm. The results are presented in Figure 4 and largely confirm one's expectations, with feasible-side convergence very close to the optimum achieved for all problems. Although the neighborhood of convergence appears to be far from the optimum in Problem P3, this is due to the geometry of the problem – because both the cost and constraint functions are close to linear in the region of the optimum, there is a wide range of points that achieve a low cost and are close to stationarity all along the constraint.
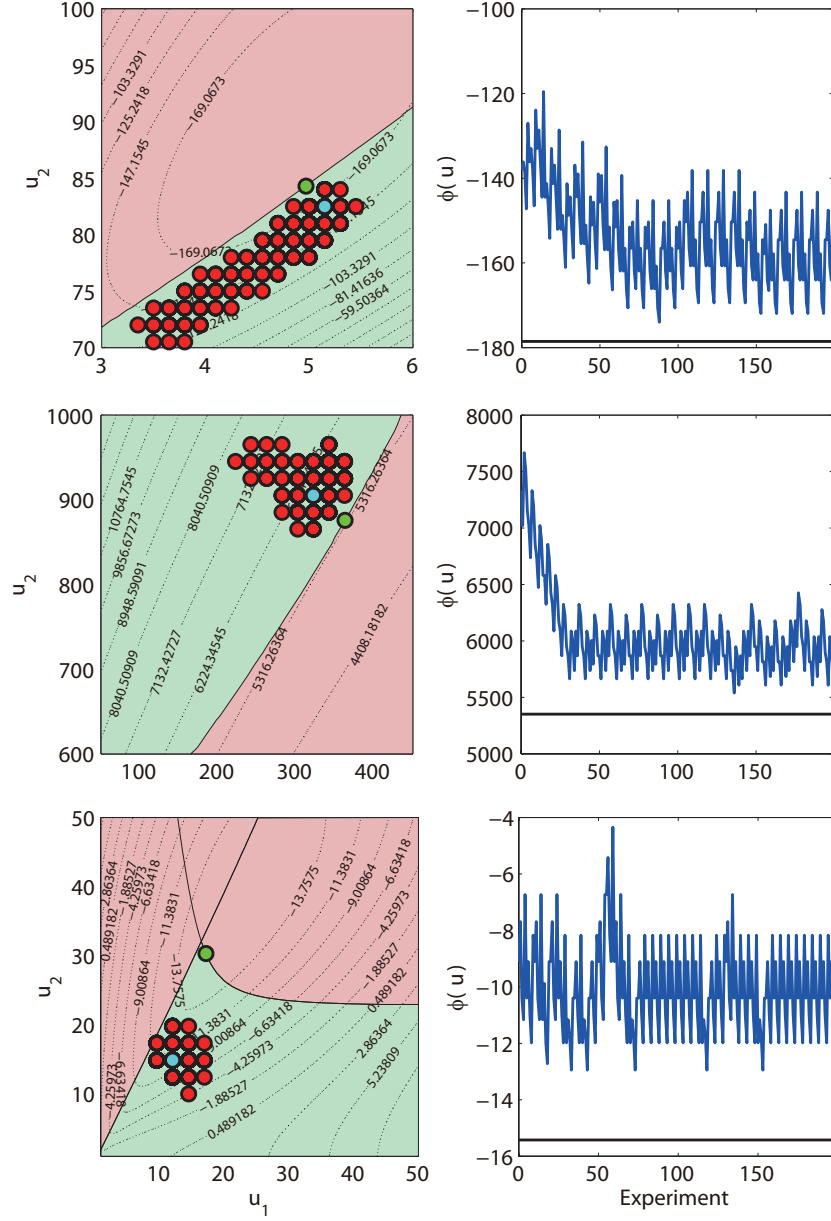
15

Figure 2: Plots of the decision variables and cost function values for Problems P2 (top), P3 (middle), and P6 (bottom). Green points in the decision-variable plots denote the true optimum, red points denote the individual experiments, and blue points denote the reference (best) point at the final tested EVOP cycle. Red regions denote the infeasible (not safe) portions of the operating space as defined by the constraints. In plotting the cost function values, the blue lines denote the true (noiseless) values, while the constant black line denotes the value at the optimum.
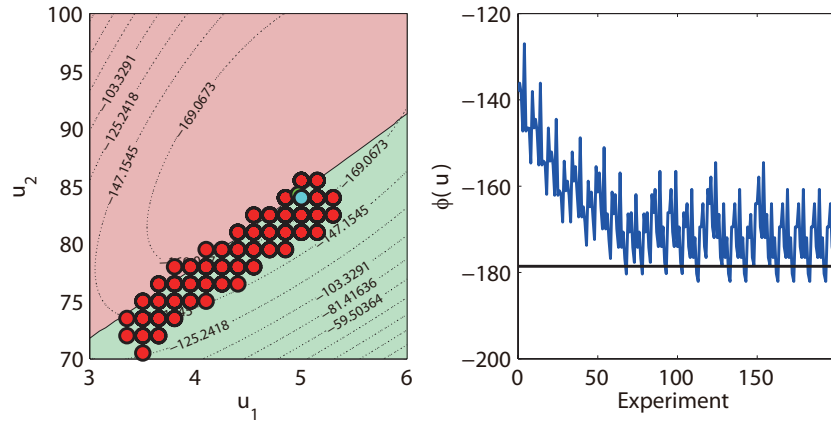
16

Figure 3: Illustration of EVOP performance for Problem P2 when the back-off is not employed.

5. Closing Remarks

This paper has contributed to the problem of obtaining information about an experimental function in the presence of general $\mathcal{C}^1$ constraints, and we have derived a rigorous constraint back-off that, when satisfied at some $\mathbf{u}^*$, guarantees that the user may perturb anywhere in the ball of radius δ_e around $\mathbf{u}^*$ without incurring constraint violations. This result is believed to constitute a useful contribution to scientific problems dealing with constrained experimental spaces, as it offers a straightforward way to guarantee safety while exploring the experimental space.

While the results simplify greatly for numerical constraints, in the case of general (likely experimental) constraints one is forced to obtain local sensitivity bounds – Lipschitz constants – for the constraint function in order to compute the appropriate back-off. To do this well may be challenging, but relevant methods do exist and may work quite well in certain contexts (Bunin & François, 2016).

The feasible-side EVOP optimization algorithm has provided an interesting application of the derived back-off result, and has been shown to work very well with respect to constraint satisfaction and convergence for three different case studies. While more involved than the traditional EVOP procedure, the algorithm nevertheless retains its ease of implementation, requiring the user to only set δ_e prior to applying it to a problem. It goes without saying that numerous performance improvements are possible but have not been the focus of this paper. One could, for example, use *all* of the measurements obtained since initialization to filter out the noise, an idea routinely employed in the SCFO solver (Bunin, 2015, §3) and recently proposed by Gao et al. (2016) in the context of multilayer real-time optimization. One could also attempt to choose search directions in a more intelligent manner, such as what is done in some derivative-free methods (Conn et al., 2009; Lewis et al., 2000). Clearly,

Figure 4: Plots of the decision variables and cost function values for Problems P2 (top), P3 (middle), and P6 (bottom) when δ_e, σ_ϕ, and $\boldsymbol{\sigma}$ are gradually reduced over the course of operation.

18

one could generalize the method to handle noise that is not white Gaussian, as well, although the computational effort may increase due to a lack of closed-form expressions.

References

Box, G. E. P. (1957). Evolutionary operation: a method for increasing industrial productivity. *J. Royal Stat. Society. Series C (Applied Statistics)*, *6*, 81–101.

Box, G. E. P. (1990). Integration of techniques in process development. *Qual. Eng.*, *3*, 9–26.

Box, G. E. P., & Draper, N. R. (1969). *Evolutionary Operation: A Statistical Method for Process Improvement*. John Wiley & Sons.

Box, G. E. P., & Wilson, K. B. (1951). On the experimental attainment of optimum conditions. *J. Royal Stat. Society. Series B (Methodological)*, *13*, 1–45.

Bunin, G. A. (2015). *The SCFO Experimental Optimization Solver: Users' Guide (version 0.91.2)*. Http://www.ccapprox.info/soft/.

Bunin, G. A. (2016). Test problem database for experimental optimization (ver. 1.3). Http://www.ccapprox.info/expopt/. Accessed March 2015.

Bunin, G. A., François, G., & Bonvin, D. (2014a). Implementation techniques for the SCFO experimental optimization framework. *arXiv:1406.3997 [math.OC]*, (pp. 1–66).

Bunin, G. A., François, G., & Bonvin, D. (2014b). On linear and quadratic Lipschitz bounds for twice continuously differentiable functions. *arXiv:1406.3991 [math.OC]*, (pp. 1–4).

Bunin, G. A., & François, G. (2016). The use of Lipschitz constants in experimental optimization. *arXiv:1603.07847 [math.OC]*, (pp. 1–41).

Bunin, G. A., François, G., & Bonvin, D. (2014c). Feasible-side global convergence in experimental optimization. *arXiv:1406.4063 [math.OC]*, (pp. 1–35).

Bunin, G. A., François, G., Srinivasan, B., & Bonvin, D. (2011). Input filter design for feasibility in constraint-adaptation schemes. In *18th World Congress of the International Federation of Automatic Control (IFAC) (Milan)* (pp. 5585–5590).

Chachuat, B., Srinivasan, B., & Bonvin, D. (2009). Adaptation strategies for real-time optimization. *Comput. Chem. Eng.*, *33*, 1557–1567.

Chen, C., & Joseph, B. (1987). On-line optimization using a two-phase approach: an application study. *Ind. Eng. Chem. Res.*, *26*, 1924–1930.

Conn, A. R., Gould, N. I. M., & Toint, P. L. (2000). *Trust-Region Methods*. SIAM.

Conn, A. R., Scheinberg, K., & Vicente, L. N. (2009). *Introduction to Derivative-Free Optimization*. Cambridge University Press.

Daniel, C., & Heerema, N. (1950). Design of experiments for most precise slope estimation or linear extrapolation. *J. Am. Statist. Assoc.*, *45*, 546–556.

Fletcher, R. (1987). *Practical Methods of Optimization*. Wiley and Sons.

François, G., & Bonvin, D. (2013). Use of convex model approximations for real-time optimization via modifier adaptation. *Ind. Eng. Chem. Res.*, *52*, 11614–11625.

François, G., Srinivasan, B., & Bonvin, D. (2005). Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty. *J. Process Control*, *15*, 701–712.

Gao, W., Wenzel, S., & Engell, S. (2016). A reliable modifier-adaptation strategy for real-time optimization. *Comput. Chem. Eng.*, *In Press*. DOI: 10.1016/j.compchemeng.2016.03.019.

Gentric, C., Pla, F., Latifi, M. A., & Corriou, J. P. (1999). Optimization and non-linear control of a batch emulsion polymerization reactor. *Chem. Eng. J.*, *75*, 31–46.

Hansen, P., Jaumard, B., & Lu, S. (1992). On using estimates of Lipschitz constants in global optimization. *J. Optim. Theory Appl.*, *75*, 195–200.

Hotelling, H. (1941). Experimental determination of the maximum of a function. *Ann. Math. Statist.*, *12*, 20–45.

Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *J. Global Optim.*, *13*, 455–492.

Kall, P., & Wallace, S. W. (1994). *Stochastic Programming*. Wiley & Sons.

Lewis, R. M., Torczon, V., & Trosset, M. W. (2000). Direct search methods: then and now. *J. Comput. Appl. Math.*, *124*, 191–207.

Li, P., Arellano-Garcia, H., & Wozny, G. (2008). Chance constrained programming approach to process optimization under uncertainty. *Comput. Chem. Eng.*, *32*, 25–45.

Marchetti, A., Chachuat, B., & Bonvin, D. (2010). A dual modifier-adaptation approach for real-time optimization. *J. Process Control*, *20*, 1027–1037.

Marchetti, A. G. (2013). A new dual modifier-adaptation approach for iterative process optimization with inaccurate models. *Comput. Chem. Eng.*, *59*, 89–100.

Montgomery, D. C. (2012). *Design and Analysis of Experiments*. Wiley.

Moré, J. J., & Wild, S. M. (2011). Estimating computational noise. *SIAM J. Sci. Comput.*, *33*, 1292–1314.

Myers, R., Montgomery, D., & Anderson-Cook, C. (2009). *Response Surface Methodology*. John Wiley & Sons.

Pfaff, G. F., Forbes, J. F., & McLellan, P. J. (2006). Generating information for real-time optimization. *Asia-Pac. J. Chem. Eng.*, *1*, 32–43.

Quelhas, A., Castro, N., & Pinto, J. (2013). Common vulnerabilities of RTO implementations in real chemical processes. *Can. J. Chem. Eng.*, *91*, 652–668.

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *Ann. Math. Statist.*, *22*, 400–407.

Rutten, K., De Baerdemaeker, J., Stoev, J., Witters, M., & De Ketelaere, B. (2015). Constrained online optimization using evolutionary operation: a case study about energy-optimal robot control. *Qual. Reliab. Engng. Int.*, *31*, 1079–1088.

Sahinidis, N. V. (2004). Optimization under uncertainty: state-of-the-art and opportunities. *Comput. Chem. Eng.*, *28*, 971–983.

Strongin, R. G. (1973). On the convergence of an algorithm for finding a global extremum. *Engin. Cybern.*, *11*, 549–555.

Torczon, V. (1997). On the convergence of pattern search algorithms. *SIAM J. Optim.*, *7*, 1–25.

Wood, G., & Zhang, B. (1996). Estimation of the Lipschitz constant of a function. *J. Global Optim.*, *8*, 91–103.

Zhang, Y., Monder, D., & Forbes, J. F. (2002). Real-time optimization under parametric uncertainty: A probability constrained approach. *J. Process Control*, *12*, 373–389.